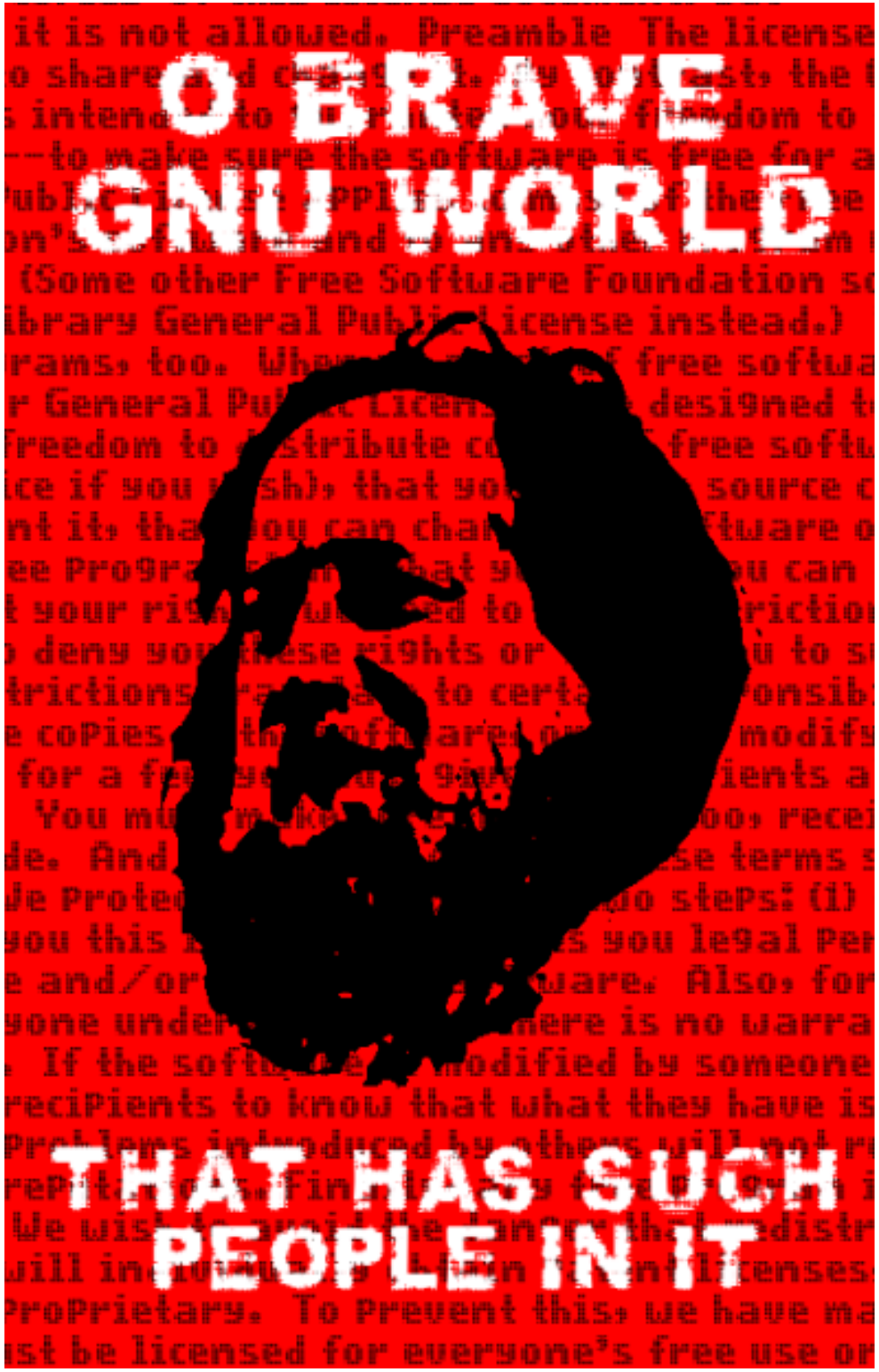
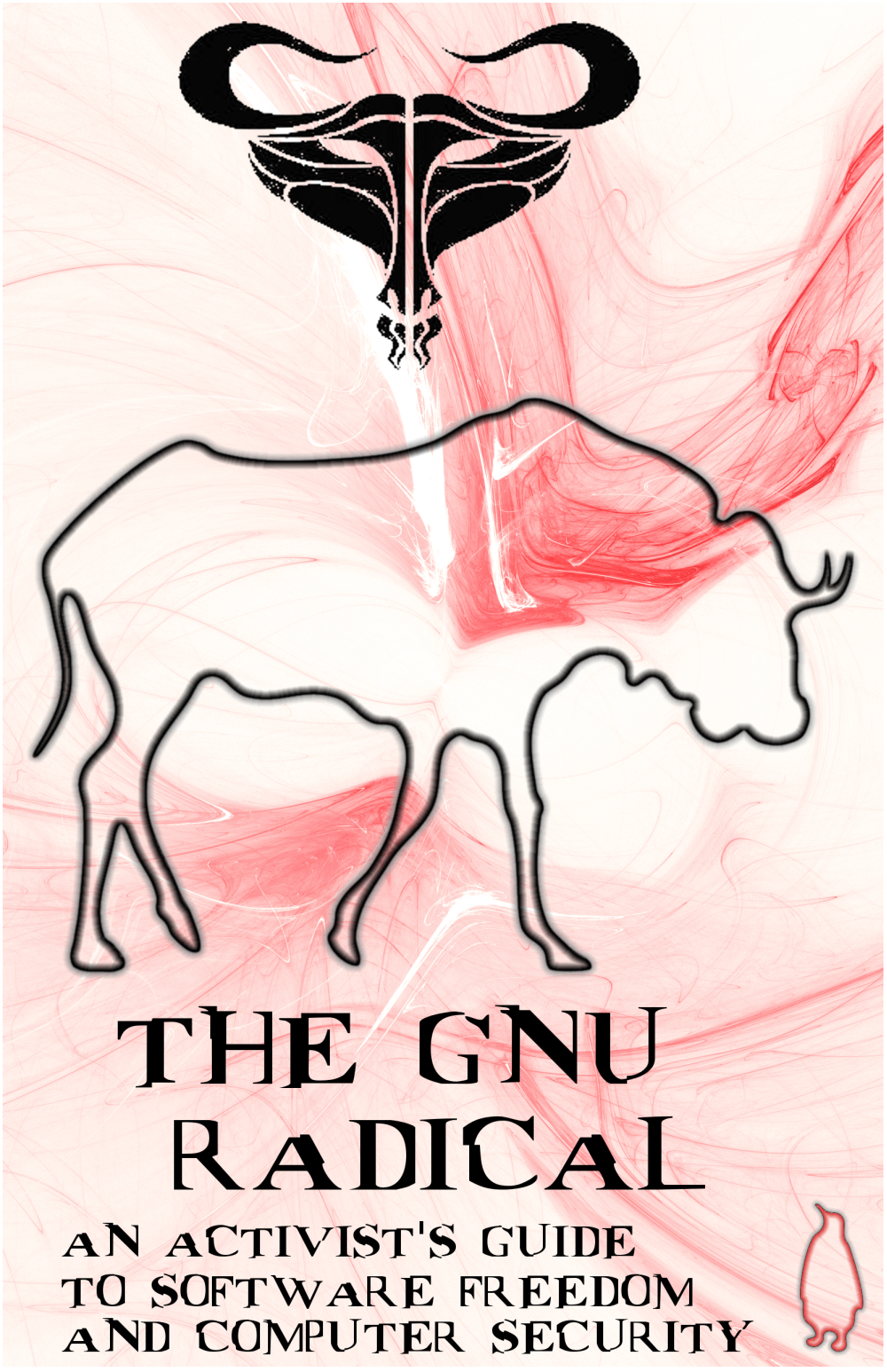

O BRAVE
GNU WORLD
THAT HAS SUCH
PEOPLE IN IT


THE GNU
RADICAL
AN ACTIVIST'S GUIDE
TO SOFTWARE FREEDOM
AND COMPUTER SECURITY

About Us

We've spent some time in the radical community as activists, organizers, and just people. As we moved through the circles of social change, we noticed one thing: though everyone had rhetoric of freedom, self-determination, and anti-coercion, they all used software that said exactly the opposite.

That's where we come in. We are the DC Radical Tech Collective. We're paranoid hackers, impassioned free software users, and advocates for social change. We've seen a void in the radical scene, and our aim is to fill it. No more will activists be able to claim ignorance of free software and privacy technology.

This zine is an introduction. It covers how to secure instant messaging and email communciation, and has an introduction to anonymity networks and anonymity in general. Also included is a reference for common GNU/Linux programs, written by a member of the ubuntu forums community, included thanks to freedom.

In the US, the NSA listens on the wires. In Europe, the EU mandates traffic logging. It's the dawn of a new year, and Joe Biden, known to privacy activsts as the Senator who tried to get crypto effectively outlawed for private citizens, is taking office as Vice President (change for America!). Activists are not going to get off easy on the electronic security front in 2009. Information about privacy and about security is more vital than ever. And we cannot trust software that does not respect our freedom to relay that information for us.

Uninstall Word. Uninstall Windows. Uninstall your chains. Embrace free operating systems like GNU/Linux and use free distributions like gNewSense, Ubuntu and Fedora. Secure your data with strong cryptography that not even the FBI can break. And don't give up your freedom.

STEAL THIS SOFTWARE

For those of us who feel that technology is a tool being used by the powers that be to control the world around us, talk of software may sound counterrevolutionary.

This thinking is false. It's what software corporations *want* you to think! What they don't want you to know is that we can control their technology better than they can! We can do it for free, and we can do it in a way that keeps cops, or feds, or whoever we don't like from reading it. Proprietary software, from the whole Microsoft complex to Internet Explorer is full of back doors through which your information seeps out and gets recorded.

To get around this issue, and issue of technological consumerism, people have been working constantly with each new form of technology to make it free and secure. When the printing press was invented, so was the radical pamphlet, known in the modern community as a zine. Pseudonyms were employed for the sake of security and new idioms were born. The foundation for a culture of encryption was built, and a language was reclaimed for the people.

When we use communications technology in the midst of a mass action either to alert dispatch that someone needs a medic or that there needs to be a last minute change in route, many scouts will encrypt their radios. Typically a frequency is easily detectable by authorities, but with the help of an additional piece of hardware, they might be able to find your channel, but they sure as shit can't hear it. We can secure communications all across the spectrum, from email to IM and beyond, and we describe how to do that in this zine.

Today we live in the communication age and that has up sides and down sides. While we've been able to share ideas, information, and stories with more speed than ever before, we're now vulnerable in new ways. The good news is that we have the power to reclaim our communication for free and secure usage by activists and create a world where our words, thoughts, and ideas will no longer be co-opted by the masters of centralized information!

Assorted Links and Resources For the Free Software Activist

http://gentoowiki.com/Main_Page
The Gentoo Wiki, while being primarily oriented towards Gentoo Gnu/Linux, is a great resource for any Gnu/Linux user. Documentation is thorough and covers a wide range of topics.

http://ubuntuforums.org/
The Ubuntu Forums are a great place to get support for Ubuntu, talk about Gnu/Linux, or holy war. Also includes a rich tutorial section that can be applied to most if not all Debian derivations.

http://www.gnewsense.org/
gNewSense is a 100% Free Gnu/Linux distribution. It is derived from Ubuntu, but removes all non-free firmware, drivers, or programs. This is ideal from an ideology and security standpoint.

http://www.gnu.org/
The Gnu Project web page has a treasure trove of philosophical essays by RMS himself, documentation for numerous Gnu Project-developed programs, and recommendations of systems and hardware that respect your freedom.

http://www.fsf.org
The Free Software Foundation is a US non-profit organization dedicated to the proliferation and defense of Free Software. While in practice the Gnu Project and the Free Software Foundation share common goals and members, the Free Software Foundation primarily concerns itself with legal or social issues. The FSF website organizes activists around FSF campaigns and maintains the Free Software Directory.

http://www.eff.org
If the FSF is the legal wing of the Gnu Project, the Electronic Frontier Foundation is the legal wing of the digital culture movement. The EFF fights legal battles for civil liberties, blogger's rights, and coder's rights, and has stood against foes including the MAFIAA, RPAA, MPAA, NSA, and others. The EFF is a vein that every activist should be tapped into if he or she wishes to keep afloat about his or her electronic rights.

A Radical's “Introduction” to Anonymity

Chapter One: Genesis

The history of humanity is the history of name struggle.

For eons we were inert masses, unable to think, unable to communicate, unable to network. One fateful day the first politician rose to power, the first state reared its head, and humanity was enslaved. But we know of that story.

What is the politician? What is the state? The state cannot be said to be something outside of fantasy, it has no arms or limbs, it cannot be touched. It is a name. And what is a politician? What is any person that can be named and called forth from the masses of humankind? Might a politician have a name as well, and with that name, whatever thoughts he might espouse to us that he may rule?

Politicians even thousands of years ago, even today, struggle against each other. But they have never struggled with ideas. They have fought for and against names.

What is an ideologue without a name? What is a person stripped of its identifier?

One day a miracle happened. All across the little globe a net was thrown, a net that carried over it information. Data. Thoughts and speech, art and science, all things. It was cyberspace, it was cipherspace, it was mindspace. And when those who understood the workings of the net grew powerful enough, they declared themselves independent. "We are not of your meatspace", they said, "Your laws of the meat world do not affect us here. We are without body and flesh, we are data. We are the information that we inhabit. You cannot stop us. You will yield to the information imperative."

In the world of ideas, the meatspace was dead. But one vestige remained. A name is nothing in mindspace. It serves only to catalog a source of ideas. But what is a source, if the ideas are sound? What possible matter is a voice to the spoken words?

Surely, the inhabitants of mindspace said, the shape of one's voice has no bearing on the matter of ones words.

And so, they stripped off the masks they had used to emulate the meat they once inhabited, they tore down the citadels of pride, they piled full the gullies of bigotry.

"This is our world now... the world of the electron and the switch, the beauty of the baud. We make use of a service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore... and you call us criminals. We seek after knowledge... and you call us criminals. We exist without skin color, without nationality, without religious bias... and you call us criminals. You build atomic bombs, you wage wars, you murder, cheat, and lie to us and try to make us believe it's for our own good, yet we're the criminals."

The Mentor
from The Hacker's Manifesto

In their new world, each idea was now free. Free to be completely independent of those which preceded it. Free to be unique of it's origin's skin color or sex preference. Free to exist and free to be shared.

We are not the future. Meatspace is not the future. Faces are not the future. Territorial turf-warring in tribal paint and over-masculine dick-measuring is not the future.

The future is mindspace. The future is ideaspace. The future is cyberspace.

The future is anonymous.

Why *NOT* to Buy Apple

Even if you aren't a free software fanatic, there's a lot of reason to dislike Microsoft right around now. Let's face it: You want a nice, shiny, fully functional, out of the box, “just works” system. You want an Apple. A lot of people are buying Apple's now. The ad campaigns from them play into the exclusive club that you'll get into when you get a Mac. The comparison to Windows boxes is clear: If you're a mac, you're a stylish, prepared, all-around-cool guy. If you're a Windows user, you're a loser. But really, you can lose just as badly with Apple.

Apple is still proprietary, non-free software: Apple has broken the interface to several free-software applications, and made it progressively harder to install your own kernel (the core system program that controls everything on a computer). These problems could be fixed easily in most cases, but in this case, only Apple can fix them, and every user is at their mercy. There's no way to put the OS X system up for peer review: the source code is locked away. And if your friend wants to try out OS X, you can't give them a copy. Those are your rights with OS X, and if you don't like them, Apple's prepared to let loose with the legal cannons.

Supporting Apple is supporting Digital Restrictions Management

The new release of the iPhone gave the world a new thing in terms of computing: a 100% locked-down system, 100% proprietary. Even Apple hardware can usually be made free by installing a free OS, like Gnu/Linux or BSD, but the iPhone 3g can't. Without the software being approved by Apple, no third-party software can be loaded on an iPhone. This isn't the first time Apple has used DRM: Every song in the iTunes store is digitally locked. DRM is used by corporations like Microsoft, Apple, and the MAFIAA (Music and Film Industry Association of America) to lock down media to enforce their monopoly in the digital age, enforcing the consumerist status quo. Don't support DRM. Don't support non-free software. Supporting both of those things is supporting the worst scenario: non-free users.

The Free Software Definition:

Software is considered free if it allows the four following freedoms:

Freedom 0: The software can be run for any purpose.

Freedom 1: The software can be modified for any purpose (*access to source code is a precondition of this freedom).*

Freedom 2: The software can be redistributed for any purpose.

Freedom 3: The modifications to the software, or modified versions of the software, can be redistributed for any purpose.

Free Software is Free as in Free Speech, Free as in Freedom, Free as in Free Society!

Chapter Two: Implications

What does it all mean?

The principles of anonymity are simple.

The basic premise of it all is that *you cannot have free speech if you can be punished for what you say.*

If you're able to be connected back to your speech, you could be harmed for it. If you espouse anti-racism, you could get killed by skinheads. If you're pro-X, you could get killed by an anti-X'er. It's just dangerous. No matter what protections there are, there's always a possibility someone could nuke your house. The only real protection is that they can't know who to nuke in the first place.

When you have truly free speech, people contribute ideas *without fear* and develop means of separating the good from the bad. *Anonymity allows a fool to be a genius once, and a genius's foolery to be discarded. Anonymity judges the content, not the metadata*.

An anonymous netizen exists without race, color, sexual preference, gender, age, or privilege. Anonymity makes it possible for bigots to work with those they hate for their metadata or for their meatspace identities.

Names promote pride and ignorance. Once you've taken a stand behind a name, you won't give it up easily. After being behind enough good ideas or enough successful events, anything you say will be automatically elevated above anyone else's. Likewise, after a few failures, your ideas will be judged not by the content of their character, but by the color of their predecessors. This allows for fully free expression to grow as a pool of knowledge is contributed to by equal peers, who then share in the collective good they've created.

It's possible to be anonymous and still be recognized as an individual, too: Just make a public key with GPG, distribute it, and sign the messages that you want to be recognized as you. Remember that you are effectively named by your key, and remember that you can have as many keys (and as many identities) as you wish. There is no reason to defend an idea like a castle: being able to admit that you're wrong without admitting to being wrong is a great asset of anonymity.

Public key systems solve the issue of trust in an anonymous network. Reputations can be built up or shattered, but even if you attain great fame as key 0x14ab3d9, you can always throw it away and start from scratch.

Anonymity allows you to be who you are without making you beholden to being you, and allows other people to accept your ideas without accepting you. There's no overhead, no metadata, no noise. Just pure information.

Chapter 3: Implementation

> *One of the defining features of anonymity is how next to impossible it is to implement.*

There are vast numbers of anonymity systems. They range from file shares to publication points to internet overlays to entirely new paradigms of networking. This chapter will go over some of the more well-known and tested ones that actually provide some guarantee of anonymity. All of the anonymity systems mentioned here are Free Software.

Consumer

Purveyors of non-free software, music, movies and other art would love for us to all be "consumers": mindless mouths gulping down their "content" and giving them as much money as possible. It offends them that people might actually ***produce*** *some of their own art, so they call us "consumers", and pass laws like "Consumer Broadband and Digital Television Promotion Act", that limits redistribution technology. We're just consumers, so why would we mind, right? Recognize that humans are capable of more than consumption by avoiding this term.*

UNIX Runlevel list:

0 - Shutdown/Halt
1 - Single-user mode (also S)
2 - Multi-user without networking
3 - Multi-user with networking
5 - Multi-user with X
6 - Reboot

> *Debian acts slightly differently from the default.*

0 - Shutdown/Halt
1 - Single-user mode
2 - Full multi-user mode (Default)
3-5 - Identical to 2
6 - Reboot

> *Manipulating the contents of the rc scripts is done via the update-rc.d program in Debian.*

update-rc.d sshd defaults - Adds sshd to the default runlevel

update-rc.d sshd start 20 2 3 4 5 . stop 20 0 1 6 - Adds sshd with explicit start/stop values

update-rc.d sshd remove - Disables sshd for all runlevels.

shutdown -h now - Shuts down and halts the system immediately. 'now' can be replaced with a number to shut down in <number> minutes.

Thanks to ajmorris of Ubuntu Forums for the original version of this guide. This has been part one of many.

Digital Rights Management

This is a term hoarders use to try to describe systems they put in place to restrict us from copying, so a better term is Digital Restrictions Management. DRM is fairly commonplace now, thanks to Apple pushing it in their music store and the MPAA putting it in DVD's so we can only watch movies on their approved devices. They don't have the right to do this, so let's not talk like they do.

ulimit -n 10240 - Sets open files limit to 10240

Changes made using the ulimit feature of bash will only exist in that shell for that session. To make commands last for each session in that shell, add the relevant commands to your shell startup file (~/.bashrc or ~/.bash_profile). To change global limits, edit /etc/security/limits.conf.

nano -w /etc/security/limits.conf

* **hard nproc 250** - Limits user processes to 250

* **hard nofile 4096000** - Limits per-process open files

Kernel limits are set via sysctl. Any changes from the default are loaded at boot-time from /etc/sysctl.conf

sysctl -a - Displays all system limits

sysctl fs.file-max - Displays max open files

sysctl fs.file-max=102400 - Sets max open files to 102400

echo "102400" > /proc/sys/fs/file-max - Does the same as above

fs.file-max=102400 - Sets parameter on boot via sysctl.conf

Runlevels

Once booted, the kernel starts init, which then starts rc, which then starts all the scripts belonging to a runlevel. The scripts are stored in /etc/rc.d/rc#.d with # being the runlevel number. The actual runlevel can be changed by signaling init. For example,

init 5 - Enters runlevel 5

<u>Trusted Computing</u>

This term is used to describe computer systems that have been backdoored or compromised, so that they only do what the manufacturer intends. An example are the controls built into Vista and the iPhone. Proprietary software already puts you at the mercy of the developers, and TC systems are designed to ensure there's no way for you to change that. This tells your computer to betray you for the sake of Hollywood megaliths, so it should be called Treacherous Computing.

Tor, The Onion Router (http://torproject.org)

Overview

Onion routing is an anonymity protocol that was one of the first to be implemented on the new internet. It works quite simply: traffic is sent from the origin to a node, which sends it to another node, which
sends it to another node, and after N nodes have been routed through, the traffic is forwarded to the destination. Tor works by encrypting traffic with AES-128 and a Public Key system, then forwarding traffic through three nodes before the traffic is decrypted and leaves the network to go to its destination. Tor was originally funded by the US Navy and is still used by the US Government for forwarding sensitive information out of potentially hostile networks.

New Terminology

Entry Node: The node at which traffic enters the network. This can be thought of as the first node in the three-node chain, or the SOCKS interface listening on your computer. While traffic is encrypted on your machine, but an Entry Node could determine your IP address (if it wasn't for Tor's protocol, but we'll get to that).

Exit Node: The node at which traffic leaves the network. The last layer of public key encryption is peeled off, the AES encryption is removed, and traffic is forwarded to the destination. *The exit node can read any of your traffic that is not encrypted from you to the destination. This includes http (web), ftp, telnet, Instant Messaging, and IRC, and of course passwords to any of these protocols. Do not send plaintext passwords over Tor!*

<u>Creator</u>

It's become popular for propagandists to refer to artists as the "creator" of a particular piece of art. The implication is that artists are gods, and have rights far beyond those of mortals. When this term is used, it's typically done to get more copyright restrictions enacted into law, which is ironic because this doesn't help the "creator": it helps the copyright holder. When you're talking about copyright restrictions, it's better to refer to the copyright holder, not the artist (most artist's don't own their work's copyright) and if you're talking about artists, remember, they're people, not gods.

Perfect Forward Secrecy: Tor has a property called "Perfect forward secrecy" for it's forwarded communication. Perfect Forward Secrecy (or PKS) means that only the last link in the chain can read any of the data. You can think of this like an encryption onion. You build up the onion on your end, wrapping an inner core of data with several layers of public key encryption. When the first node gets the onion, it peels one layer, and the next node does the same, until the exit node is reached and the data is forwarded.

Bad Node or Bad Exit: An exit node that takes advantage of its link in the chain to sniff data. A famous bad exit was able to sniff email passwords for thousands of government embassy logins. Bad exits might sniff your data, or they might modify it to insert advertisements or hostile data to break your anonymity.

Strengths

Tor's strength comes from its uniformity. At any point in the chain besides the exit, no node knows where in the chain it is. This means that encrypted traffic from you into the entry node and from the entry node into a circuit node is just traffic, and its origin can't be determined. Tor protects against forms of *Traffic Analysis*, an attack on anonymity that involves watching connections. If an adversary could see all the connections of all the Tor nodes in the world, they could break Tor. But since there are Tor nodes all over the world, in various countries with various diplomatic status between them, that won't happen. Previously it was thought that Tor would be trivial to break due to the low number of nodes, but since then Tor has grown from having 400 nodes to 5000 nodes, with an average of 1000 online at any given time. *To strengthen your anonymity and everyone else's, run a Tor node.* Not only will this help the network, it'll make your anonymity stronger, as traffic coming from you could be originated from you OR forwarded by you.

Tor is also low-latency. While it might not be low-latency compared to your normal net connection, it is certainly low latency compared to other anonymity systems, like Freenet or GNUnet. This is both a strength and a weakness, as it can make some timing attacks easier.

Commands for user information

id - Shows active user ID with login and group

last - Shows last logins on the system

who - Shows who is logged on the system

who -b - Shows last boot time

groupadd admin - Creates the group 'admin'

useradd -c "Joe Bloggs" -g admin -m joe - Creates user Joe Bloggs and adds him to the group admin.

usermod -a -G <group> <user> - Adds existing user to group

userdel Joe - Deletes the user 'Joe'

Limits

Some applications require higher limits on open files and sockets. The default limits are sometimes too low.

ulimit -a - Displays limits

ulimit -H -a - Displays hard limits (set at boot from /etc/security/limits.conf)

ulimit -n - Displays open files limit

Intellectual Property

As far as doublethink goes, this is one of the worst offenders. For starters, an idea shouldn't be anyone's "property": that's like saying gravity is Newton's "property", or relativity is Einsteins "property". If an idea has the capacity to help humanity, then it can't be locked away. But this term is used by propagandists to make us think that thoughts, programs, movies, music, and art are property and should be locked away. It lets restriction advocates call sharing "stealing" and "theft". This term is by definition a catch-all for the legal fields of copyright, patents, and trademarks, which evolved differently and for different purposes. Just call them for what they are, instead of using the propaganda buzzword.

last reboot- show system reboot history

lsb_release -a- show full release info of any LSB distribution

cat /etc/debian_version- Get Debian version for debian derived distros

Information about your Hardware

Kernel detected hardware

dmesg - Prints detected hardware and boot messages

lsdev - information about installed hardware

dd if=/dev/mem bs=1k skip=768 count=256 2>/dev/null | strings -n 8 - Reads BIOS strings

lspci - Shows PCI devices

lsusb - Shows USB devices

dmidecode - Shows DMI/SMBIOS information (hardware information from BIOS)

lshal - Show a list of all devices with their properties

lshw - Shows a list of connected hardware

cat /proc/cpuinfo - Print information about your CPU

cat /proc/meminfo - Print information about hardware memory

grep MemTotal /proc/meminfo - Display your physical memory

watch -n1 'less /proc/interrupts' - Watch changeable interrupts continuously. 'watch -n1' is very useful to watch any changing file.

free -m - Prints used and free memory in megabytes

cat /proc/devices - Displays configured devices

One feature is possibly the biggest advantage of using Tor: Hidden Services. Hidden Services are just like any other service on the net: IRC servers, websites, shell servers, chat servers, anything that runs on TCP (and most of the net runs on TCP), but with one important difference. Hidden Services are anonymous. With normal websites, you can always find the owner, and possibly persecute/prosecute him for his speech, but with a Hidden Service, she's hidden behind Tor. Plus, even plaintext content is safe, because ALL traffic is encrypted end-to-end with a Hidden Service.

Hidden Services are like a whole new internet. There's a culture on the ones that are open to the public of anonymity and free information. The author of this article was inspired many times by events or statements on the Hidden Service scene, and without the environment it provided, might not have written this article. That being said, hidden services allow many that are persecuted to engage in behavior that many in society find utterly disgusting. True freedom isn't for the faint of heart.

Weaknesses

While this isn't a weakness of Tor (there is no way to implement this in any system ever), the biggest drawback of Tor is the lack of trust in the node operators. While this won't compromise anonymity, it can compromise data. While using Tor, make sure to take the same precautions as you would on any other untrusted network. Encrypt everything. Passwords should be sent in SSL or secure hashed form, messages should be encrypted. While bad nodes on Tor aren't nearly as prevalent as good ones, there is no way to know if an exit node is sniffing your traffic.

Tor is also vulnerable to a few classic attacks on anonymity networks, including the "Giant Overseer" attack and timing/correlation attacks. The Giant Overseer attack is simple: If the adversary can see all traffic on all nodes of the Tor network, the game is over. But this attack isn't really feasible unless the Illuminati (exists and) wants to break Tor, or if one government took over the entire world.
A more potent attack is a timing attack: If I watch Bob sending a request

for a file, and then observe Alice getting a request of equal size, followed by Alice sending a 300MB file, if Bob gets a 300MBfile, there is a good chance it might be Bob talking to Alice. This could be defeated with padding (making all data distributed on the network use a certain amount of data all the time), but that would be impractical and severely impact Tor's speed. This attack would be very useful in discovering the location of a Hidden Service, but it would take a very large amount of resources to successfully complete.

Although this attack is impractical due to the US Navy's endorsement and the US Government's (and other governments) widespread use of Tor, Tor is extremely vulnerable to attacks on centralized resources. Tor nodes look up hidden service and node addresses via a centralized directory, and while the directory is mirrored, only a few servers are "authoritative" and have supreme say over the network.

Closing Thoughts

Tor has taken a lot of flak from people who are pissed about the ability of exit node opers to sniff data, but it should be kept in mind that sniffing data or the potential to sniff data does not compromise anonymity. Encrypted traffic is the only truly safe content when using Tor to access public-web servers. The Hidden Service feature, a main focus of Tor development, is a great boon to radicals, and in fact, Tor has become home to ALF communiques from all around the world and Chinese dissident speech. Certain Indymedia's also run a Tor portal, allowing users to have hidden-service level anonymity, but communicate with those that don't. Another site that allows this is masked.name, a blogging/publishing site hosted by a prominent Tor Citizen, or Torizen. Tor, like any system, should not be trusted 100%. However, it can be safely used for any variety of things that would be impossible otherwise. It cannot be stressed enough that a Tor user is only as good as their configuration: Tor can be broken via client-side holes in a variety of ways. But with a safe configuration and a cautious end-user, Tor can not only be safe, but possibly the safest means of anonymous TCP traffic.

Introduction to Gnu/Linux System: Common Programs

This is a list of commonly used programs on Gnu/Linux systems, categorized by function, with short descriptions. This is based primarily on this post on Ubuntu Forums: http://ubuntuforums.org/showthread.php?t=842307

System Information

These programs display system statistics, messages, and load information.

top - displays and updates the top cpu processes

mpstat 1 - displays the first processor's statistics, in Debian/Ubuntu repositories as package 'sysstat'.

vmstat 2 - displays virtual memory statistics

iostat 2- displays I/O statistics at 2sec. intervals. In Debian/Ubuntu repositories as package 'sysstat'.

tail -n 500 /var/log/syslog - prints the last 500 messages logged via the syslog interface.

tail -n 500 /var/log/messages - prints the last 500 kernel messages.

tail /var/log/warn - prints last system warnings, see/etc/syslog.conf for details.

uname -a - get kernel version, OS, CPU arch, and other information

uptime- show how long the system has been running along with average CPU load

hostname - displays the system's hostname

hostname -i- displays the system's IP address

man hier - file system hierarchy manual page

Avoiding Doublethink

One of the side effects of the technological revolution is that we're constantly within reach of communications, and as a side effect of that, we're constantly within the reach of the mass media. We're bombarded with advertisements, “news”, and even less veiled propaganda. And because the people behind this carpet bombing of our minds are those with money, the purveyors of non-free software and non-free computing in general are typically one of the loudest speakers.

As a result of this, the parts of our culture, even the culture of free software to an extent, that deal with things these propagandists dislike are *filled* with doublethink. We're told that downloading a song or giving a friend a CD is piracy – equating sharing music with attacking an unarmed ship on the high seas. In movie theaters, we're told that if we “steal” a film by sharing it, we're in effect “stealing” the car of whoever was behind the movie, and putting thousands of common working-class people out of work (even though they're paid regardless of how much the movie makes in theaters!).

We face this doublethink in free software, in free societies, and anywhere the interests of the old guard of capitalists have their market dominance threatened by emerging technology that returns the freedom back to the people. Throughout this zine, there are examples of this doublethink, and suggestions of more appropriate terms that more accurately describe what's going on, as well as some assorted facts that clarify the current situation of copyright, trademarks, and patents. After you get this knowledge, don't let it rot in your head – help your friends and community reject doublethink!

i2p (http://i2p2.de)

Overview

In a lot of ways, i2p is the opposite of Tor. i2p is written in Java, Tor is written in C. Tor uses TCP for its transport (and can only transmit TCP streams), i2p uses UDP for transport (and can transport UDP and TCP streams), i2p was designed originally for two-way anonymity (in the style of hidden services), Tor was designed as an outproxy system. The differences between the two networks offer an intriguing opportunity to compare implementation of the same general goal.

New Terminology

Eepsite: The analogue of a Hidden Service in the i2p world, an Eepsite is a website that is only accessible on i2p.

Eeproxy: An Eeproxy is part of the middleman system that allows i2p to communicate with TCP-using protocols. An Eeproxy specifically deals with HTTP, and is used by a web browser to access .i2p sites.

Garlic Routing: Similar to Onion Routing, the major difference of Garlic Routing is the inclusion of other data between layers of encryption. This partially defends i2p against timing attacks, as data within the encrypted payload is not necessarily just the data received.

Tunnel: In i2p, every node has set of inbound and outbound tunnels. These tunnels are the tendrils through which i2p is able to communicate with the outside world anonymously. The design of i2p, with each node having inbound and outbound tunnels, also means that every i2p node is anonymous.

Strengths

i2p's design is more of a replacement for the IP layer or an IP overlay than a TCP overlay such as Tor. In this way, i2p is more diverse and possibly more resilient than Tor, as UDP applications are able to utilize it nativity, and TCP applications can be coerced through a TCP stream layer.

Another focus of i2p is decentralization. There are few central points of weakness in the i2p system: unlike Tor, which bootstraps nodes from a central directory, i2p has a distributed database which it uses for lookups, bootstraps itself off of a distributed system, and even holds the source code in a distributed framework.

Another design strength of i2p is the fact that all participants are fully anonymous. i2p lacks mass outproxy support, and in effect the network functions as a fully anonymous internet, running on an anonymous IP implementation. This, combined with i2p's variable-length chains, allows for a large amount of diversity in usage. Modified clients or other projects exist to provide distributed forums, jabber servers, IRC servers, email, and even high-bandwidth p2p such as BitTorrent.

Weaknesses

i2p is vulnerable to the some of the same attacks as Tor, with the exception of timing attacks, due to garlic routing. i2p is powerless against an observer who can watch every node in the network, and i2p is also weak against brute-force denial of service attacks more so than Tor, due to it's Java implementation.

At this point, your system will generate random noise to use for the keypair. Let this run for a while. If you're on a Gnu or BSD system (this includes OS X), you can put load on the hard disk or bash on the keyboard to generate entropy. i2p has not received the peer review or attention as Tor, so developer error could be a possible factor. i2p's use as an IP overlay is especially important. Currently to the author's knowledge, i2p is the only system that will enable anonymizing UDP applications.

Closing Thought

i2p is a great anonymity system for those that are willing to make certain sacrifices, mostly in speed. Java is not a fast platform, and i2p knows this. However, i2p has many benefits that make it possibly more resilient to attack than Tor. Anyone who needs anonymity should not play systems bigot, but instead familiarize themselves with everything that might help them, and i2p certainly will.

It's safe now to upload your key to a keyserver. To do this, whip out your favorite text editor and edit your gpg.conf file. This file will be in ~/.gnupg/ on Gnu/BSD/UNIX systems, and in C:\Documents and Settings\<username>\Application Data\gnupg\ on a Windows NT system. Add these lines to it:

x-hkp://pgp.mit.edu/
x-hkp://keyserver.noreply.org

Then run the command *gpg --send-keys*. This will upload your public key to the keyservers listed, and from those keyservers, it will propagate out to the rest of the world.

Now that you have your key, go into Thunderbird and install Enigmail by going to Tools>Addons, and then selecting install. Restart Thunderbird after installing Enigmail.

After you restart, there will be a new menu entry: OpenPGP. If this menu entry isn't there, Enigmail wasn't successfully installed, and you should see what when wrong. If it is, go to Account Settings, and select OpenPGP Security under your account. Check the box marked "Enable OpenPGP for this identity", and make sure that Enigmail will use the correct key. I suggest also setting Enigmail to sign all messages by default, and selecting the "Send OpenPGP Key ID" option, as this will allow people with email clients that support it to automatically download and verify your signed messages.

At this point, test out your configuration by writing a new email to a friend that has PGP already. You should be able to sign (OpenPGP>Sign Message) and encrypt (OpenPGP>Encrypt Message) your message. Send a signed and encrypted message to a friend or to yourself to see if everything has worked. You should have to put in the password for your key to encrypt and decrypt the message.

Congrats, you now have secure email. Remember, your security is only as good as your habits! NEVER risk your private key falling into hostile hands, and if it does, be fast with your revocation certificates.

Happy hacking!

The first step we need to take is generating our key pair. This will be our public/private key that we'll use to sign our emails, and that others will use to encrypt messages sent to you. This can be done in two ways.

The first method is using the command line and invoking gpg with the --gen-key command, like this: "*gpg --gen-key*". This will bring up an interactive dialog. For most options, feel free to use the default. However, for the option "*What keysize do you want?*", enter 4096, the maximum. This will make your key as secure as possible at the expense of some time every time you use it. This is well worth it.

What key size do you want? (2048) 4096

When it comes time to put in your password, Make sure you use a secure passphrase with uppercase characters, lowercase characters, numbers, and symbols. Make your password more than 16 characters long. Passwords are often the weakest point in an encryption scheme. Make sure that isn't the case here.

Congrats, you have a key pair! The first thing we're going to do with it is generate a revocation signature, so that in case our keypair ever goes out of our control, we can nuke it out of existence. You can do this with the gpg command: *--gen-revoke.*
--gen-revoke takes one argument, the user ID of the key to generate a revocation certificate for. This can be the name or email address attached to the key you just made.

gpg --gen-revoke alice@host.com

gpg will then ask you why you're generating a revocation certificate. It doesn't matter what you put here, and it's fine to go with *0 = No reason specified*. Generate the certificate, then store it in a safe place. Back it up on a CD, and hide the CD. If an adversary obtains this certificate, they can use it to render your key useless. If you forget the passphrase to your key or lose your private key, you can use it to let everyone know that your key isn't yours anymore.

Freenet (http://freenetproject.org)

Overview
One similarity in Tor and i2p is that both are low-latency forwarding systems. Tor is a TCP overlay, and i2p is an IP overlay. Freenet is completely different.

Freenet could be appropriately called a publishing system. It is possibly the most resilient publishing system ever created. Freenet is designed to stave off censorship, by providing distributed storage and anonymity. Once a file is uploaded to Freenet, it is nearly impossible to remove. Freenet is, of course, anonymous, and is capable of operating in an even-more-anonymous "Darknet" mode. While Tor or i2p are good for IM, IRC or Email, Freenet is hands down the best for communiques, information on opponents, or anything that must not go down.

New Terminology
Opennet: This is the mode that Freenet operated under in version 0.5 and is an optional mode of operation in Freenet 0.7. This is a method for a Freenet node to discover other Freenet nodes, and does so openly - hence the name. In Opennet mode, a node will connect first to "seed" nodes, which then offer the node connections to other nodes, and so on. This is vulnerable to attack more so than ...
Darknet: This is the opposite of Opennet. Instead of connecting to any possible peer, a node is configured to only connect to trusted peers. Don't be too stingy or liberal with your definition of trust - peers you connect to still don't know if connections originate from or are forwarded by you, and if you have enough people as peers that also have enough people as peers, "small-world routing" can be used to create a highly efficient network.

Strengths

Freenet is possibly the most reliable way to publish data. Once put on the network, data cannot be manually deleted by any single party, and will only be removed after very long periods of disuse and want for space on the part of other, more highly used files. This means that unless your content isn't downloaded in many years, it won't disappear. No other system can claim this to the extent which Freenet can. In addition to its reliability, with enough people in Darknet mode, a small-world network will be formed allowing for easy routability. Small world networking refers to the principle that there is a small number of hops between any two given acquaintances: a cyberpunk version of "six degrees of Kevin Bacon". While seeming fantastic, this works well in practice. The main barrier to its implementation is that people who use Freenet quite often don't know many others who also use it.

Freenet is one of the longer-running anonymity systems, and has seen a lot of development and use over the years. As a result, many possible holes have been covered, and countermeasures have been devised to a number of attacks. Freenet is highly fault-tolerant. If a hostile user tried to DoS a node by requesting lots of data, the data would eventually be cached by the node immediately next to them in the chain. Freenet is highly distributed, and anyone using Freenet also operates a Freenet node, forwarding traffic and storing data for the rest of the network. It is impossible to determine what content is hosted on your node. The cloud is able to move data around to where it's needed most, so if Bob and Alice both lived in the same area, and downloaded a file multiple times, eventually a node in that area would cache the content, allowing for lower latency and further decentralization.

Open Source (pt. I)

Most of us have been “brought up” thinking of open source as a god thing. While it certainly is, as activists, it isn't the best thing. Open Source is primarily a development method, while Free Software is a social movement. The Open Source movement, while in practice working towards the same ends as the Free Software movement, doesn't value freedom, but values pragmatic performance. As activists, we know that freedom is more important; that if a program enslaves it's users, it isn't “better” even if it's faster or lighter. Just by saying Free Software instead of Open Source, you emphasize freedom.

So how do you join and get in on this sweet game of trust relationships, public/private key pairs, and digital signature? Well!
Here's how to set up a key pair and an email client so that you can do what all of us really want to do: encrypt and sign emails!

This assumes that you have Thunderbird as a mail client, configured correctly. This is easy to do on Gmail: the instruction are linked from the configuration page. Check with your email provider for specific instructions on how to set up Thunderbird for POP (or IMAP) and SMTP access.

The two programs you need to download are GnuPG and the Enigmail Thunderbird addon. If you don't have Thunderbird, you'll need to install that as well.

For Gnu or BSD systems, you can typically install Thunderbird, Enigmail and GnuPG from your package manager. Damn, that's easy, isn't it?

Windows users will have to get Thunderbird:
http://www.mozilla.com/en-US/thunderbird/,
Enigmail :
http://enigmail.mozdev.org/download/index.php,
and GnuPG:
http://www.gnupg.org/download/index.en.html.

Once these are all installed, we can get started setting up the crypto.

A Public Key Infrastructure (or PKI) is a system that doesn't solve much relating to the strength of crypto, but solves immense problems relating to the distribution of keys. Suppose Alice wants to send an encrypted email to Bob. Using conventional methods, Alice needs some way to communicate the encryption key to Bob. In a lot of situations, this isn't feasible. A PKI solves this problem by having two keys: a public key and a private key. A public key is intended to be widely distributed and can be used to encrypt a message or file. But whatever is encrypted with the public key can only be decrypted by the private key. Likewise, the private key can be used to "sign" a message or file, and the public key can verify that signature, allowing Alice not only to encrypt a message to Bob, but include verification that the message is in fact from Alice.

There's only one problem here. How does Alice know the key she has for Bob is actually Bob's? How does Bob know the signature on the message from "Alice" is actually from her? The best way to get a key is in person, but if you could always meet people in person, there would be no need for a PKI. To solve this, a system called the "trust web" was created. Suppose that while Alice has never met Bob, their mutual friend Cathy is able to verify that the public keys that they both have are genuine. If Alice trusts Cathy, then she can use Bob's key without worry.

This trust web is implemented via key-servers and key signing. Trusted key servers hold repositories of public keys that are available for download, and those keys can be signed by people that have verified the identity of the purported keyholder. **Key verification should always be done in person. If you do not know a person, or if they do not have sufficient identification, do not verify their key**.

Weaknesses

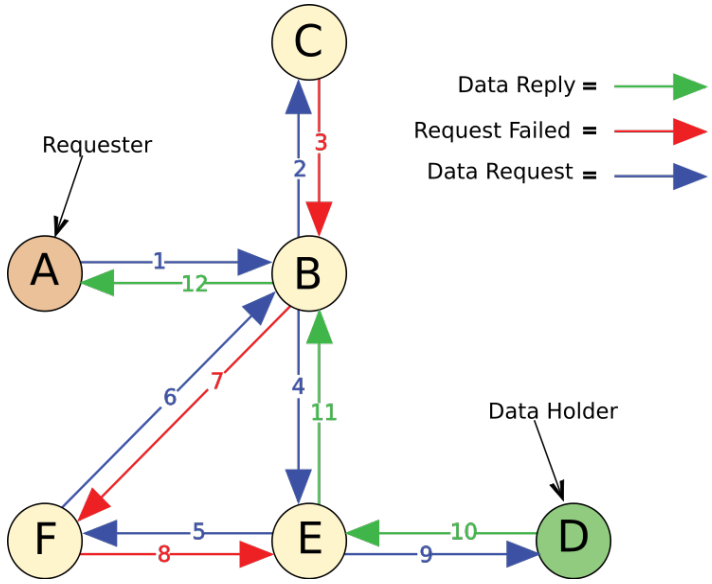
A major weakness in Freenet is the discrepancy in security between the Darknet and Opennet modes. While Darknet is far more secure, it is harder to implement in practice. Opennet provides a far easier solution, but allows the network to be attacked by hostile ISPs or governments. There are a large amount of possible attacks on Opennet, almost as much as there are on the rest of the Freenet structure.

Freenet, like i2p, is also implemented in Java. Up until recently, there was no free software version of the Java Virtual Machine, so java code could be potentially untrusted. Currently, Freenet is not compatible with the OpenJDK, so this problem remains.

Final Thought

Freenet, while often overlooked due to high latencies and demands on the end-user, is a thoroughly reliant system of disseminating information, especially information that is disliked by powerful entities. Freenet's network structure and design are resilient, and the developers have experience dealing with anonymity attacks. Freenet was so good that the Chinese government blocked version 0.5. That should say more than I can about it's potency.

A Freenet Data Request



GNUnet (http://gnunet.org)

Overview

GNUnet, the Gnu Project's filesharing protocol, is a relative newcomer on the anonymity scene. Unlike Tor or i2p or Freenet, GNUnet's goal is to be a peer-to-peer protocol for sharing information freely. It draws on the Gnu Project's standards of modularity and portability to produce a powerful application, but it is still in its infancy, both in terms of code maturity and network growth.

New Terminology

Transport: A transport is a means by which GNUnet uses its network. Currently, GNUnet has four transports: TCP, UDP, HTTP, and SMTP. Transports are fully modular, and have various strengths and weaknesses. For instance, the SMTP transport, while high latency and abuse-vulnerable, is able to get around just about any firewall or NAT (since everywhere allows email).

Strengths

GNUnet's biggest strength is its modularity. The GNUnet application operates as a client and server, and a GNUnet server can serve to multiple clients. Currently, clients exist for basic command-lines, the GTK toolkit, and the QT toolkit. GNUnet is also modular in it's transport layer, allowing users behind restrictive firewalls to still have access to the network.

Weaknesses

As great as the GNUnet codebase is, it's a project that just needs more love. GNUnet is relatively immature, and has a large number of bugs. Any anonymous network needs to grow a bit before becoming fully usable, and GNUnet isn't quite there yet. Hopefully, it will be soon.

Final Thought

GNUnet is a great application with a moderately powerful network that isn't used or developed anywhere near as much as it should be. That's just about all that needs saying.

Secure (encrypted) Email using Mozilla Thunderbird, Enigmail, and GnuPG

I've asked radicals before when I hear them mention email, "How are you securing yourself?" Almost invariably the answer I receive is a puzzled look or "We know not to use email for anything incriminating."

Nothing incriminating, hm? What might that mean? You won't talk to people who are less security-conscious than you, and who might let something slip in an email? Remember, if one of your friends is flagged by a keyword-based monitoring system, that brings attention to anyone and everyone that person is in contact with. You've never implied in an email that you're an anarchist, a radical, or any kind of dissenter at all?

I would hazard to guess that 99% of you at some point have admitted something to a few things you might be regretting now. You might remember programs like CARNIVORE and wonder if you're on a watch-list because of an email now yourself. The FBI and the NSA have the hardware and the time to make nice big pretty webs of all our little scenes, and chart out who's talking to who via email, instant messenger (though now they shouldn't be able to listen in on *those* conversations ;)

The assumption I make (and you might want to make too) is that the enemies we as activists have are watching us talk over email and IM, and that they're making neat little charts and networks, just like they do to the Mafia. We might not be able to stop them from knowing who's talking to who, but we can definitely stop them from knowing what we're talking about, whether that's an action, a relationship, or something totally benign. We can make it outright impossible for them to read our emails. Even if they could brute-force through it (which they probably can't) they couldn't do that to <u>everything</u>. So spread this to all of your friends, put links to resources in your email signatures, spread crypto. One person with crypto sticks out and draws attention, everyone using crypto sends a giant middle finger to the NSA. We're going to be securing email with GnuPG, a free software implementation of PGP, which is an implementation of an encryption scheme known as a Public Key infrastructure.

Fourth, change keys, or subkeys, fairly often. Every year should be okay. Too often is annoying to everyone trying to contact you, and doesn't give you much benefit, but 1 - 4 times per year shouldn't be a huge hassle to anyone, and gives a good deal of security. Using subkeys for gpg is a good idea if you use it to communicate with many people (this should be everyone), since you don't have to have everyone verify again when you switch subkeys, unlike if you switch keys.

Fifth, for any really important or sensitive data, use more than just full disk encryption. I like truecrypt because of the hidden volume option, but there is nothing wrong with using encfs or even just encrypting the important files with gpg. The important thing is to have a step that requires you to think before having access to sensitive data, and to put one more barrier between an attacker and the data that really matters. It isn't unthinkable that someone could somehow get ahold of your computer when it is logged in, so it is wise to make sure that even in that event, all is not lost.

This is a very, very long way from exhaustive, but I hope it gives you a good starting point, or, if you already know your way around this stuff, a good reminder, on general computer security. Before I finish though, I promised a bit on limitations.

The best algorithms in the world backed up with a hundred character random passphrase doesn't mean anything if someone else knows it. If some person or persons have decided in advance that they need to break your encryption (rather than, for example, stealing/confiscating your computer and then coming up against the security), they will try to get the passphrases from you by hook or by crook. Attacking strong cryptography directly is not feasible. Never tell anyone your passphrase, and avoid talking about your passphrase. Never write your passphrase down. Make sure no one shoulder surfs your passphrase. As a corollary, it's not unheard of to plant cameras to pick up a passphrase; don't let it happen to you. Think about using keyfiles as well as a passphrase. Be careful, stay safe.

Conclusion

We live in the dawning of a golden new era, where information will be traded freely and those that seek to harm the flow of that information will not be tolerated. The principles and implementations of anonymity that stand today will march forward into this golden dawn, carrying with them the hopes of humanity.

I am not my name. I am not my clothes. I am not my skin color, gender, or sex preference. I am not my BMI or my breast size. I am not my meat.

I am anonymous. And in my anonymity, I am part of something greater than I. I am part of the next stage of humanity, where names will count for nothing.

No gods. No masters. No borders. No nations. No names.

Just humanity. Just ideas.

INSTANT MESSAGING SECURITY

While many radicals seem to have embraced the falsity that “The internet is always insecure and should not be used", a hell of a lot of them still end up using instant messaging. This is quite natural and should be expected; even the most security-conscious (but non-technical) radical is still just a person and should be expected to use communication methods.

However, this approach lends itself to further insecurity. While the eavesdropper may not know of your specific plans, they might still be privy to things the attacker probably should not know; such as romantic troubles, sources of stress, objects of affection, times when you'll be "unavailable" to others, what your political views even are, etc.. Thankfully, IM can be secured trivially, and this tutorial will attempt to concisely offer how to do just that.

This tutorial will rely on the LibPurple family of instant messaging clients, for the following reasons:

- They are the easiest to implement encryption on
- They are fully cross-platform
- They are relatively secure

For Windows, we'll be installing the Pidgin instant messaging client, and then extending it with the OTR plugin for encryption. Pidgin supports AIM, MSN, YIM, IRC, XMPP, and many other protocols. To install pidgin, download the installer and go through it. You can get the Pidgin installer at http://pidgin.im. The installer should be intuitive for any Windows user; configure the locations and components and let loose. When you're done, head over to:
http://www.cypherpunks.ca/otr/#downloadscyphterpunks.ca/otr
and grab the windows installer:
http://www.cypherpunks.ca/otr/binaries/windows/pidgin-otr-3.1.0-1.exe
for the OTR plugin. Run that, and remember, it shouldn't create any shortcuts. It's a plugin, not a program.

I'm sure we can all think of a few situations where we would rather not have our computers be an open book to anyone who cares to look. Thankfully, securing your computer against an attacker with a great deal of resources is not very difficult Here are some guidelines for general security. They aren't really in any order, and all of them are really rather important.

First, use free, open source software exclusively on any computer that will touch anything you want to keep safe. A system running GNU/Linux or a BSD variant is a must. On that machine, try not to run anything closed source. The problem with closed source software is that you don't actually know what it's doing, and so it categorically can't be trusted. An obvious example is Microsoft's well documented backdoors for law enforcement. All the hard work that you do to secure your computer will be for naught if the proprietary software you run is backdoored or phones home with compromising data. If you have no unix experience, I suggest using the latest Ubuntu alternate install CD, and to install with the free software only option using dm-crypt/LUKS full disk encryption. This isn't a how to, so I won't go in depth, but having all your data encrypted and on a system you can trust is the first step. There are pages that could be written on just FDE, but I won't be the one to write them, at least not right now. Just know that this part is crucial.

Second, use good passphrases. The passphrase is the weak point in the encryption link. The best passphrases don't have words, contain numbers and symbols, and are at *very least* 16 characters long (though personally, I tend to use at least 30. Use the longest passphrases you can actually remember). If you have trouble remembering that sort of thing, there are alternatives. Though I stand by the random string approach, something like diceware.com is a good start. Perhaps use 7+ words and inject a couple random numbers or symbols for good measure.

Third, secure your communications. There are articles on OTR, GPG, and tor in here. Read them, follow them, enjoy the privacy.

bring the companies into the NSA monitoring fold. This relationship was exposed by a few brave whistle-blowers years later, after the NSA had set up massive wiretapping operations on some of the "backbone" wires of the Internet. The actions of these whistle-blowers came to effectively nothing; no congressional action was taken and while the wiretapping program was shut down, it is very safe to assume that the NSA's budget for such acts will not at all shrink and that they will be back.

AUSTRALIA

Australia is perhaps the worst "free" nation to inhabit in terms of censorship. Australia currently is implementing a wide-spread censorship regime that will suppress access to wide ranges of content, ranging from "child pornography" (like Wikipedia) to "hacking" and "terrorist" content. While the initial trials in implementing the system have been failures, the plans are ongoing and uninterrupted.

EU Links:

https://secure.wikimedia.org/wikinews/en/wiki/Data_Retention_Directive_passed_by_EU_Parliament
http://www.theregister.co.uk/2005/12/14/eu_data_retention_vote/
http://www.privacyinternational.org/article.shtml?cmd[347]=x-347-496240

Germany Links:

http://blogoscoped.com/archive/2007-11-12-n32.html
http://www.vorratsdatenspeicherung.de/content/view/209/1/lang,en/
http://www.kreativrauschen.com/blog/2008/06/04/data-retention-effectively-changes-the-behavior-of-citizens-in-germany/

UK Links:

http://community.zdnet.co.uk/blog/0,1000000567,10009938o-2000331777b,00.htm?new_comment
http://en.wikinews.org/wiki/UK_ISPs_erect_%27Great_Firewall_of_Britain%27_to_censor_Wikipedia
http://www.edri.org/edrigram/number6.16/uk-data-retention

US Links:

http://www.eff.org/nsa/faq
https://secure.wikimedia.org/wikipedia/en/wiki/NSA_warrantless_surveillance_controversy
https://secure.wikimedia.org/wikipedia/en/wiki/NSA_call_database
https://secure.wikimedia.org/wikipedia/en/wiki/Mark_Klein
http://blog.wired.com/27bstroke6/2008/12/ny-times-nsa-wh.html

Australia Links:

https://secure.wikimedia.org/wikipedia/en/wiki/Internet_censorship_in_Australia
http://www.efa.org.au/Issues/Censor/cens1.html

Now fire up Pidgin. It should prompt you to create an account, so add accounts for your protocol(s) of choice. Remember, if you select the "Remember Password" option, your password will be stored in a **cleartext** file on your hard drive. Make sure it is in a secure partition or folder.

Click the Tools tab, then click the Plugins menu. Scroll down to the entry for "Off-the-Record Messaging". Check the box next to the plugin to enable it. This will cause a grey icon to appear in the lower left of any message windows you have open. The icon is of the former logo of Pidgin being eavesdropped on by others. If the person you're chatting with has the capability (meaning they have one of the setups found in this how to or a compatible one), clicking on this icon will allow you to transition to Off-the-Record messaging.

If you use Apple's OS X, this whole process is even easier. Just install Adium, available from http://www.adiumx.com/. The encryption subsystems are included. Adium uses LibPurple, so it provides mostly the same functionality as Pidgin, with the exception of a few protocols that the Adium developers thought weren't suited for an instant messaging client.

If you're running GNU/Linux, it's even easier! Simply install Pidgin and Pidgin-otr from your distribution's repositories. On Ubuntu, both of them are installed by default. Enable the plugin after installation and go. If your distribution does not provide packages, download the .deb or .rpm package from the Debian or Fedora repositories. ***Remember to install from the repositories. It is unnecessary to build from source.***

For Free/Free, Open source software (pt. II)

English is in many ways a confusing language, and one of those ways is the ambiguity of certain words. While most Romance languages have two words for the terms “zero cost” and “liberated”, English doesn't. We just have one word: free. Free software is often distributed at zero cost, but the thing that makes it free software is the fact that it has freedom and allows the user to maintain their freedom. By saying “Free, Open Source Software”, you're de-emphasizing the freedom, the most important part of any program. It's sufficient to just say “Free Software”, possibly with the explanation as to which free you mean. To show that freedom isn't just a marketing buzzword, translate “free software” to your own language, and say that. “Libre software” is a good example.

After setting up Pidgin and OTR, remember, both parties must have the software to establish encrypted communications. Remember to verify keys (OTR will ask you to do this at first, write down the fingerprints and verify them in person) with the person you're communicating with, to make absolutely sure that you're talking to who you think you are, all the time.

OTR is a crypto scheme specifically tailored to instant messaging. While initially, parties exchange key “fingerprints”, that can be used to verify who you're talking to. This is why you must verify keys in person: the fingerprint is to an OTR'd conversation what seeing the person's face is to a real-life one. However, after that exchange, **no** message afterwards is in any way verified with that fingerprint, allowing you to deny sending any given message. You can see who you're talking to, but an eavesdropper can't tell what you're saying, or even if you're saying it!

Think you don't need crypto? Read this article (http://en.wikipedia.org/wiki/Room_641A) and see if it changes your mind. And if you're still not convinced, set up the crypto anyways; it takes all of five minutes and it'll help out your friends that also have it.

A Gnu using OTR

Looking Forward -- A Glimpse into the GNU Year

Privacy has been steadily and consistantly attacked by the state as long as citizens have had the ability to protect their communications with encryption the state could not crack. This article outlines the moves Governments around the world are making in 2009 to inhibit, restrict, or utterly deny privacy to their subjects.

EUROPE

A few years ago, the European Parliament passed Directive 2006/24/EC. That act mandated member states to implement data holding practices in order to trace the identity of citizens on the internet. States that are implementing the measure in 2009 include Germany and the UK. While Germany's measures are being met with heavy citizen resistance, it should be noted that the Germany measures are limited to logging who occupies what IP (Internet Protocol) address at any given time, and is comparable to what American ISPs already practice on a wide scale.

The UK's measures have been said to be significantly more Orwellian. Already (it should be noted this is NOT a result of the EU mandate) 95% of UK ISP's filter internet traffic based on the findings of the Internet Watch Foundation, a non-governmental (for or non?)-profit organization that attempts to flag websites as hosting child pornography. The technology behind this filtering is about the same as what China uses to censor its citizens: traffic is shunted through a "filtering proxy", which removes content deemed by the IWF to be unsatisfactory. The existence of this blocking came to public light only after Wikipedia came under the IWF's gun within the last few months.

The ruling party of Sweden, in absence of any EU directives, voted to monitor all traffic crossing Swedish borders in an attempt to detect "terrorist" activity. Sweden has long been a safe hostel for popular BitTorrent tracker PirateBay, but mounting pressure from American megaliths such as the RIAA and MPAA (and their joint project, the MAFIAA) has driven the Swedish government to extremes previously not foreseen.

USA

America leaves behind an eight-year stretch of unprecedented denial of basic civil liberties. Under the Bush Government, US citizens were logged, traced, and tapped like never before. However, this increase cannot be 100% attributed to the trusted fail-safe of the Bush Government, '9/11 and terrorism', as documents have emerged showing the NSA (the leading agency in Orwellian surveillance) initiating relationships with telecommunications companies earlier than 9/11 with intent to